

CURRENT LISTING OF THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1        1. (Previously Presented) A method performed by software embodied in a  
2 computer-readable storage medium and executed by a computer in a database system that stores  
3 a join view associated with plural base relations, the method comprising:

4              receiving modification operations that modify at least two of the base relations of  
5 the join view, wherein the at least two base relations comprise a first base relation and a second  
6 base relation;

7              performing partitioning of the received modification operations by submitting at  
8 least some of the modification operations operating on the first base relation to a first session,  
9 and submitting at least another of the modification operations that operate on the second base  
10 relation to a second session;

11              grouping the at least some of the modification operations in the first session  
12 operating on the first base relation into a first transaction,

13              wherein the at least another modification operation in the second session is part of  
14 a second transaction; and

15              schedule the transactions to avoid execution of modification operations of more  
16 than one of the at least two base relations at one time in the database system.

1        2. (Previously Presented) The method of claim 1, wherein scheduling the  
2 transactions comprises:

3              determining that the first transaction conflicts with the second transaction based  
4 on the first and second transactions modifying more than one base relation of the join view; and

5              selecting one of the first and second transactions for execution in the database  
6 system.

1        3. (Previously Presented) The method of claim 2, wherein selecting one of the first  
2 and second transactions comprises selecting the first transaction,  
3              the method further comprising storing the second transaction in a queue.

1           4. (Previously Presented) The method of claim 3, further comprising waiting for the  
2 first transaction to complete execution before scheduling the second transaction for execution.

1           5. – 8. (Cancelled)

1           9. (Previously Presented) The method of claim 1, further comprising:  
2               storing pending transactions in plural queues corresponding to respective plural  
3 sessions of the database system; and  
4               selecting one of the pending transactions from the queues to schedule for  
5 execution in the database system based on whether the one pending transaction conflicts with  
6 one or more executing transactions in the database system.

1           10. (Previously Presented) The method of claim 9, further comprising determining  
2 that the one pending transaction conflicts with the one or more executing transactions in response  
3 to determining that the one pending transaction modifies a different one of the base relations of  
4 the join view than a base relation of the join view modified by an executing transaction.

1           11. (Previously Presented) The method of claim 9, further comprising applying a  
2 technique to prevent starvation of a particular one of the pending transactions in response to  
3 determining that the particular one pending transaction has been in one of the queues for longer  
4 than a predetermined time period.

1        12. (Previously Presented) An article comprising at least one computer-readable

2 storage medium containing instructions that when executed cause a computer to:

3              receive modification operations that modify at least two of the base relations of a  
4 join view, wherein the at least two base relations comprise a first base relation and a second base  
5 relation;

6              perform partitioning of the received modification operations by submitting at least  
7 some of the modification operations operating on the first base relation to a first session, and  
8 submitting at least another of the modification operations that operate on a second base relation  
9 to a second session;

10             group the at least some of the modification operations in the first session  
11 operating on the first base relation into a first transaction,

12             wherein the at least another modification operation in the second session is part of  
13 a second transaction; and

14             schedule the transactions to avoid concurrent execution of transactions of more  
15 than one of the at least two base relations of the join view.

1        13. (Previously Presented) The article of claim 12, wherein scheduling the  
2 transactions comprises:

3              determining that the first transaction conflicts with the second transaction based  
4 on the first and second transactions modifying more than one base relation of the join view; and  
5              selecting one of the first and second transactions for execution in the database  
6 system.

1        14. (Previously Presented) The article of claim 13, wherein selecting one of the first  
2 and second transactions comprises selecting the first transaction,

3              the instructions when executed causing the system to further store the second  
4 transaction in a queue.

1        15. (Previously Presented) The article of claim 14, wherein the instructions when  
2 executed cause the computer to wait for the first transaction to complete execution before  
3 scheduling the second transaction for execution.

1           16. – 17. (Cancelled)

1           18. (Previously Presented) The article of claim 12, wherein the instructions when  
2       executed cause the computer to:

3                 in response to a particular one of the modification operations to modify one of the  
4       base relations, place an exclusive lock on the one base relation, and place a predefined lock on  
5       the join view,

6                 the predefined lock conflicting with each of a shared lock and an exclusive lock  
7       placed on the join view, but the predefined lock not conflicting with another predefined lock  
8       placed on the join view.

1           19. (Cancelled)

1           20. (Previously Presented) The first system of claim 22, wherein the controller is  
2       adapted to identify the modification operations on the second base relation as conflicting with the  
3       modification operations on the first base relation in response to determining that the modification  
4       operations on the second base relation are modifying a different base relation of the join view  
5       than the modification operations on the first base relation.

1           21. (Previously Presented) The first system of claim 22, wherein the controller is  
2       adapted to open plural sessions with the database system that is separate from the first system,  
3                 the controller to further:

4                 identify modification operations on the first base relation that modify  
5       distinct portions of the first base relation; and

6                 submit the identified modification operations that modify distinct portions  
7       of the first base relation through different sessions for concurrent execution in the database  
8       system.

1        22. (Previously Presented) A first system comprising:

2              a controller having one or more processors to:

3                  receive modification operations to modify plural base relations of a join

4 view, the modification operations comprising modification operations to modify a first base

5 relation of the join view, and modification operations to modify a second base relation of the join

6 view; and

7                  re-order the received modification operations to avoid concurrent

8 execution of modification operations of more than one of the plural base relations of the join

9 view,

10                 the re-ordering to cause modification operations on the first base relation

11 of the join view to be scheduled for execution, and to cause modification operations on the

12 second base relation to be queued for execution after completion of the modification operations

13 on the first base relation,

14                 wherein certain of the modification operations on the first base relation

15 comprise modification operations of a set of one or more tuples of the first base relation, and

16 wherein the controller is adapted to:

17                 group the modification operations on the set of one or more tuples of the

18 first base relation into a transaction; and

19                 submit the transaction to a database system separate from the first system

20 for execution.

1        23. (Previously Presented) The first system of claim 22, wherein the controller

2 comprises a load utility to submit the modification operations to the database system.

1        24. (Previously Presented) The first system of claim 23, wherein the load utility

2 comprises a continuous load utility.

1        25. (Previously Presented) The first system of claim 23, wherein the load utility

2 comprises a first load utility, and the controller comprises a second load utility to concurrently

3 submit other modification operations to the database system.

1            26. (Previously Presented) The first system of claim 25, further comprising plural  
2 platforms on which corresponding first and second load utilities are executable.